24. (newly entered) The processing unit of Claim 22, wherein said upper level cache is a store-through cache.

25. (newly entered) The processing unit of Claim 22, further comprising:

means for loading a cache line including the requested value into said upper level cache in response to a determination that it would be efficient to load the cache line into said upper level cache.

26. (newly entered) The processing unit of Claim 22, wherein said cache controller selects said victim cache block based at least in part on the cache misses of said lower level cache.

27. (newly entered) The processing unit of Claim 22, and further comprising means for selecting a victim cache block in said upper level cache for receiving the requested value based at least in part on the cache activity of said lower level cache.

REMARKS

This Amendment is submitted in response to the Office Action dated December 19, 2000, having a shortened statutory period set to expire March 19, 2001. In the present Amendment, Applicant has proposed amendments to the specification to provide serial numbers of cross-referenced applications. In addition, Applicant has amended Claims 1, 10-11 and 21, canceled Claims 8-9 and 19-20, and has entered Claims 22-27. Thus, Claims 1-9, 12-18 and 21-27 are currently pending.

In paragraph 5 of the present Office Action, Claims 1-21 are rejected under 35 U.S.C. § 102(b) as anticipated by U.S. Patent No. 5,737,751 to *Patel et al.* (*Patel*). That rejection is respectfully traversed, and favorable reconsideration of the claims is requested.

Applicant believes that *Patel* does not render the present claims unpatentable because *Patel* does not identically disclose each feature of the present claims. As amended herein, Claim 1 (and similarly Claims 11 and 22) recites a "method of operating a multi-level cache of a computer system," which following a miss in both the upper and lower level caches, "select[s] a victim cache block in the lower level cache for receiving the requested value based at least in part on cache hits in the upper level cache." Thus, according to the present invention, a victim entry in a lower level (e.g., L2) cache is selected for replacement at least in part utilizing upper level (e.g., L1) cache hit information. As will be appreciated, this recitation stands in direct contrast to the conventional multilevel cache hierarchies in which only L1 misses (not hits) are made visible to lower level caches.

At page 4, paragraph 5 of the present Office Action, the Examiner generally cites Figure 3 of *Patel* as teaching "selecting a victim cache block in the lower level cache for receiving the requested value based at least in part on cache hits in the upper level cache." However, Figure 3 of *Patel* does not disclose or discuss victim selection. Instead, *Patel's* Figure 3 teaches that if a load request misses in both the L1 and L2 caches, a linefill is performed in both the L1 and L2 caches. If, on the other hand, a store request misses in both the L1 and L2 caches, a linefill is performed in the L1 cache and not the L2 cache. Although such linefill operations may require selection of a victim cache block, *Patel* clearly fails to discuss selection of the victim cache block and certainly does not disclose selection of a victim in a lower level cache utilizing cache hit information for an upper level cache.

As should thus be apparent, *Patel* does not render independent Claims 1, 11 and 22 or their respective dependent claims unpatentable because *Patel* does not show or suggest each feature of the present claims. Applicant therefore believes that all pending claims are in condition for

allowance and respectfully requests such allowance.

Please charge IBM Corporation Deposit Account No. **09-0447** in the amount of $36.00 for two additional claims in excess of 20. No additional fee is believed to be required; however, in the event any additional fees are required, please charge IBM Corporation Deposit Account No. **09-0447**.

Respectfully submitted,

Brian F. Russell
Reg. No. 40,796
BRACEWELL & PATTERSON, L.L.P.
Suite 350 Lakewood on the Park
7600B North Capital of Texas Highway
Austin, Texas 78731
(512) 343-6116

ATTORNEY FOR APPLICANTS

SPECIFICATION

The paragraph appearing at page 1 has been amended as follows:

The present invention is related to the following applications filed concurrently with this application: U.S. Patent Application Serial No. [09/_____] 09/340,077 entitled "QUEUE-LESS AND STATE-LESS LAYERED LOCAL DATA CACHE MECHANISM" (attorney docket no. AT9-98-778); U.S. Patent Application Serial No. [09/_____] 09/340,076 entitled "LAYERED LOCAL CACHE MECHANISM WITH SPLIT REGISTER LOAD BUS AND CACHE LOAD BUS" (attorney docket no. AT9-98-779); U.S. Patent Application Serial No. [09/_____] 09/340,075 entitled "LAYERED LOCAL CACHE WITH IMPRECISE RELOAD MECHANISM" (attorney docket no. AT9-98-780); U.S. Patent Application Serial No. [09/_____] 09/340,073 entitled "METHOD FOR UPPER LEVEL CACHE VICTIM SELECTION MANAGEMENT BY A LOWER LEVEL CACHE" (attorney docket no. AT9-98-782); U.S. Patent Application Serial No. [09/_____] 09/340,082 entitled "LAYERED LOCAL CACHE WITH LOWER LEVEL CACHE UPDATING UPPER AND LOWER LEVEL CACHE DIRECTORIES" (attorney docket no. AT9-98-783); U.S. Patent Application Serial No. [09/_____] 09/340,078 entitled "HIGH PERFORMANCE STORE INSTRUCTION MANAGEMENT VIA IMPRECISE LOCAL CACHE UPDATE MECHANISM" (attorney docket no. AT9-98-784); U.S. Patent Application Serial No. [09/_____] 09/340, 079 entitled "HIGH PERFORMANCE LOAD INSTRUCTION MANAGEMENT VIA SYSTEM BUS WITH EXPLICIT REGISTER LOAD AND/OR CACHE RELOAD PROTOCOLS" (attorney docket no. AT9-98-785); U.S. Patent Application Serial No. [09/_____] 09/340,080 entitled "METHOD FOR LAYERING LOCAL INSTRUCTION CACHE MANAGEMENT" (attorney docket no. AT9-98-786); and U.S. Patent Application Serial No. [09/_____] 09/340,081 entitled "METHOD FOR LAYERING LOCAL TRANSLATION CACHE MANAGEMENT" (attorney docket no. AT9-98-787).

CLAIMS

1. (amended) A method of operating a multi-level cache of a computer system, comprising the steps of:

monitoring cache activity of an upper level cache and a lower level cache both associated with a processor of the computer system, said monitoring including monitoring cache hits in the upper level cache;

issuing a request from the processor to load a value, wherein the request misses the upper level cache and the lower level cache; and

selecting a victim cache block in the lower level cache for receiving the requested value based at least in part on [prior cache activity of] cache hits in the upper level cache.

2. (unchanged) The method of Claim 1 wherein the victim cache block is further selected based in part on the cache activity of the lower level cache.

3. (unchanged) The method of Claim 1 wherein said selecting step takes place out of a critical path of execution of a core of the processor.

4. (unchanged) The method of Claim 1 wherein said issuing step issues a request to load operand data.

5. (unchanged) The method of Claim 1 wherein said selecting step includes the step of identifying a less recently used cache block in the lower level cache.

6. (unchanged) The method of Claim 1 further comprising the steps of:

returning the requested value to the processor;

determining that it would be efficient to currently load into the upper level cache a cache line which includes the requested value; and

in response to said determining step, loading the cache line into the upper level cache.

7. (unchanged) The method of Claim 1 wherein:

said monitoring step monitors cache misses of the upper level and lower level caches; and

said selecting step selects the victim cache block based at least in part on the cache misses of the lower level cache.

8. (canceled)

9. (canceled)

10. (amended) The method of Claim 1 [8] further comprising the step of selecting a victim cache block in the upper level cache for receiving the requested value based at least in part on the cache activity of the lower level cache.

11. (amended) A computer system comprising:

a system memory device;

means for processing program instructions;

means, connected to said processing means, for caching values stored in said system memory device, said caching means having at least an upper level cache and a lower level cache both associated with said processing means;

means for monitoring cache activity of said upper level cache and said lower level cache including cache hits in the upper level cache; and

means for selecting a victim cache block in said lower level cache for receiving a value

specified in a load request issued by said processing means, wherein the load request missed said upper level cache and said lower level cache, based at least in part on [the cache activity of] cache hits in said upper level cache.

12. (unchanged) The computer system of Claim 11 wherein said selecting means is located out of a critical path of execution of a core of said processing means.

13. (unchanged) The computer system of Claim 11 wherein said upper level cache is an operand data cache.

14. (unchanged) The computer system of Claim 11 wherein said selecting means identifies a less recently used cache block in said upper level cache.

15. (unchanged) The computer system of Claim 11 wherein:

said upper level cache is an L1 cache; and

said lower level cache is an L2 cache.

16. (unchanged) The computer system of Claim 11 wherein said upper level cache is a store-through cache.

17. (unchanged) The computer system of Claim 11 further comprising:

means for returning the requested value to said processing means in response to the load request missing said upper level cache; and

means for loading a cache line which includes the requested value into said upper level cache in response to a determination that it would be efficient to currently load the cache line into said upper level cache.

18. (unchanged) The computer system of Claim 11 wherein:

said monitoring means monitors cache misses of said lower level cache; and

said selecting means selects said victim cache block based at least in part on the cache misses of said lower level cache.

19. (canceled)

20. (canceled)

21. (amended) The computer system of Claim 11 [19] further comprising means for selecting a victim cache block in said upper level cache for receiving the requested value based at least in part on the cache activity of said lower level cache.

22. (newly entered) A processing unit, comprising:

at least one instruction execution unit;

at least an upper level cache and a lower level cache;

a cache controller that, responsive to receipt of a load request issued by said at least one execution unit that missed said upper level cache and said lower level cache, selects a victim cache block in said lower level cache for receiving a value specified in the load request, wherein the selection is based at least in part on cache hits in said upper level cache.

23. (newly entered) The processing unit of Claim 22, wherein said upper level cache is an operand data cache.

24. (newly entered) The processing unit of Claim 22, wherein said upper level cache is a store-through cache.

25. (newly entered) The processing unit of Claim 22, further comprising:

means for loading a cache line including the requested value into said upper level cache in response to a determination that it would be efficient to load the cache line into said upper level cache.

26. (newly entered) The processing unit of Claim 22, wherein said cache controller selects said victim cache block based at least in part on the cache misses of said lower level cache.

27. (newly entered) The processing unit of Claim 22, and further comprising means for selecting a victim cache block in said upper level cache for receiving the requested value based at least in part on the cache activity of said lower level cache.